AMENDMENTS TO THE CLAIMS

Please amend the claims as follows.

1. (Previously Presented) A method for obtaining data from a kernel, comprising:

   obtaining data from the kernel; and

   storing the data in a data set in an aggregation buffer using an aggregation function.

2. (Previously Presented) The method of claim 1, wherein the data set comprises a key component, an aggregation identifier component, and a value component.

3. (Original) The method of claim 1, wherein obtaining data comprises:

   obtaining an expression, a new value, and an aggregation identifier, and

   generating a key using the expression and the aggregation identifier.

4. (Original) The method of claim 3, wherein storing the data set comprises:

   storing the key in the key component,

   storing the aggregation identifier in the aggregation identifier component, and

   updating a current value in the value component using the new value and the aggregation function.

5. (Original) The method of claim 4, wherein storing the data set further comprises:

generating a hash key using the key;

searching for a hash bucket corresponding to the key using the hash key;

searching for a hash chain element in the hash bucket corresponding to the key;

updating the value component of the data set associated with the hash chain element if a hash chain element corresponding to the key is found, wherein the updating the value component comprises applying the aggregation function to the current value in the value component using the new value as input;

creating a new hash chain element if the hash chain element corresponding to the key is not found, wherein creating a new hash chain element comprises associating a new data set with the new hash chain element, storing the key in a key component of the new data set, storing the aggregation identifier in an aggregation identifier component of the new data set, and storing an initial value in a value component of the new data set; and

updating the value component associated with the new hash chain element, wherein the updating the value component associated with the new hash chain element comprises applying the aggregation function to the initial value using the new value as input.

6. (Original) A method for storing data in a data set, wherein the data set comprises a key component, an aggregation identifier component, and a value component, comprising:

obtaining an expression, a new value, and an aggregation identifier;

generating a key using the expression and the aggregation identifier; and

storing the data set in a buffer, wherein storing the data set comprises storing the key in the key component, storing the aggregation identifier in the aggregation identifier component, and updating a current value in the value component using the new value and an aggregation function.

7. (Original) The method of claim 6, wherein storing the data set comprises:

generating a hash key using the key;

searching for a hash bucket corresponding to the key using the hash key;

searching for a hash chain element in the hash bucket corresponding to the key;

updating the value component of the data set associated with the hash chain element if a hash chain element corresponding to the key is found, wherein the updating the value component comprises applying the aggregation function to the current value in the value component using the new value as input;

creating a new hash chain element if the hash chain element corresponding to the key is not found, wherein creating a new hash chain element comprises associating a new data set with the new hash chain element, storing the key in a key component of the new data set, storing an aggregation identifier in the aggregation identifier component of the new data set, and storing an initial value in a value component of the new data set; and

updating the value component associated with the new hash chain element, wherein the updating the value component associated with the new hash chain element comprises applying the aggregation function to the initial value using the new value as input.

8. (Original) The method of claim 7, wherein the hash chain element is associated with the data set using a pointer.

9. (Original) The method of claim 7, wherein the new hash chain element is associated with the new data set using a pointer.

10. (Original) The method of claim 6, wherein the expression comprises an n-tuple.

11. (Cancelled)

12. (Currently Amended) A method for integrating data into a user-level table, comprising:

obtaining a data set from an aggregation buffer, wherein the data set comprises a key component, an aggregation identifier component, and a value component;

obtaining an aggregation identifier matching a value of the aggregation identifier in the aggregation identifier component of the data set to obtain a user-level table key;

hashing the user-level table key to obtain a generated hash key;

updating a value component of a user-level table entry if a user-level table entry having a hash key matching the generated hash key is found, wherein updating the value component comprises applying an aggregation function corresponding to the aggregation identifier to the value component using the new value as input;

creating a new user-level table entry if a user-level table entry having a hash key matching the generated hash key is not found, wherein creating the new user-level table entry comprises storing the generated hash value, and an initial value in a value component of the new user-level table entry in the new user-level table entry; and

updating the value component in the new user-level table entry, wherein updating the value component in the new user-level table entry comprises applying the aggregation function corresponding to the aggregation identifier to the value component in the new user-level table entry using the value component of the data set as input.

13. (Original) The method of claim 12, wherein the user-level table is a hash table.

14. (Previously Presented) The method of claim 12, wherein obtaining the aggregation identifier matching the value of the aggregation identifier comprises searching at least one selected from the group consisting of a user-level dictionary and a kernel level dictionary.

15. (Original) A data aggregation buffer comprising:

> a data set, wherein the data set comprises a key component for storing a key, an aggregation identifier component for storing an aggregation identifier component, and a value component for storing a value;
>
> a hash chain referencing the key component of the data set using a pointer; and
>
> a hash bucket organizing the hash chain.

16. (Original) The data aggregation buffer of claim 15, wherein the key is generated from a function using an aggregation identifier.

17. (Original) The data aggregation buffer of claim 15, wherein the key is generated from a function using an aggregation identifier and an expression.

18. (Cancelled)

19. (Original) A data aggregation system comprising:

> an aggregation buffer associated with a central processing unit;
>
> a user-level buffer operatively connected to the aggregation buffer and arranged to copy a data set from the aggregation buffer; and
>
> a user-level hash table storing a portion of the data set from the user-level buffer using a user-level dictionary,
>
> wherein the user-level dictionary provides a reference into the user-level hash table.

20. (Original) The data aggregation system of claim 19, wherein the data aggregation buffer comprises:

> a data set, wherein the data set comprises a key component for storing a key, an aggregation identifier component for storing an aggregation identifier component, and a value component for storing a value;
>
> a hash chain referencing the key component of the data set using a pointer; and

a hash bucket organizing the hash chain.

21. (Original) The data aggregation system of claim 20, wherein the key is generated from a function using an aggregation identifier.

22. (Original) The data aggregation system of claim 20, wherein the key is generated from a function using an aggregation identifier and an expression.

23. (Original) An apparatus for obtaining data from a kernel, comprising:

    means for obtaining data from a kernel;

    a tracing framework; and

    means for storing the data in a data set in the tracing framework,

    wherein the tracing framework comprises:

        an aggregation buffer associated with a central processing unit;

        a user-level buffer operatively connected to the aggregation buffer copying the data set from the aggregation buffer; and

        a user-level hash table storing a portion of the data set from the user-level buffer using a user-level dictionary,

        wherein the user-level dictionary provides a reference into the user-level hash table.

24. (Currently Amended) A computer system on a network obtaining data from a kernel comprising:

    a processor;

    a memory;

    a storage device; and

    software instructions stored in the memory for enabling the computer system to:

        obtain[[ing]] data from the kernel; and

        store storing the data in a data set an aggregation buffer using an aggregation function.